In [25]:
```
import json
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


with open('receipts.json') as f:
    receipts_data = [json.loads(line) for line in f]


receipts_list = []
items_list = []


unique_item_attributes = [
    'barcode', 'brandCode', 'competitiveProduct', 'competitorRewardsGrou
    'discountedItemPrice', 'finalPrice', 'itemNumber', 'itemPrice', 'meta
    'needsFetchReviewReason', 'originalFinalPrice', 'originalMetaBriteBa
    'originalMetaBriteItemPrice', 'originalMetaBriteQuantityPurchased',
    'pointsEarned', 'pointsNotAwardedReason', 'pointsPayerId', 'preventT
    'quantityPurchased', 'rewardsGroup', 'rewardsProductPartnerId', 'tar
    'userFlaggedDescription', 'userFlaggedNewItem', 'userFlaggedPrice',
]

for receipt in receipts_data:
    receipt_id = receipt.get('_id', {}).get('$oid', None)
    if receipt_id:
        receipt_data = {
            'receipt_id': receipt_id,
            'bonusPointsEarned': receipt.get('bonusPointsEarned', None),
            'bonusPointsEarnedReason': receipt.get('bonusPointsEarnedRea
            'createDate': receipt.get('createDate', {}).get('$date', Non
            'dateScanned': receipt.get('dateScanned', {}).get('$date', N
            'finishedDate': receipt.get('finishedDate', {}).get('$date',
            'modifyDate': receipt.get('modifyDate', {}).get('$date', Non
            'pointsAwardedDate': receipt.get('pointsAwardedDate', {}).ge
            'pointsEarned': receipt.get('pointsEarned', None),
            'purchaseDate': receipt.get('purchaseDate', {}).get('$date',
            'purchasedItemCount': receipt.get('purchasedItemCount', None
            'rewardsReceiptStatus': receipt.get('rewardsReceiptStatus',
            'totalSpent': receipt.get('totalSpent', None),
            'userId': receipt.get('userId', None)
        }
        receipts_list.append(receipt_data)

        if 'rewardsReceiptItemList' in receipt:
            for item in receipt['rewardsReceiptItemList']:
                item_data = {'receipt_id': receipt_id}
                for attr in unique_item_attributes:
                    item_data[attr] = item.get(attr, None)
                items_list.append(item_data)

# Converting lists to DataFrames
receipts_df = pd.DataFrame(receipts_list)
items_df = pd.DataFrame(items_list)

# Convert Unix timestamps to datetime objects in receipts DataFrame
```

```
date_columns = ['createDate', 'dateScanned', 'finishedDate', 'modifyDate
for col in date_columns:
    receipts_df[col] = pd.to_datetime(receipts_df[col], unit='ms', error
```

In [26]:
```
print(items_df.shape, receipts_df.shape)
```

```
(6941, 35) (1119, 14)
```

In [ ]:

Type *Markdown* and LaTeX: α^2

In [38]:
```
missing_values_receipts_percentage = (receipts_df.isnull().sum() / len(re


missing_values_items_percentage = (items_df.isnull().sum() / len(items_d

missing_values_receipts = receipts_df.isnull().sum()
missing_values_items = items_df.isnull().sum()



missing_values_receipts_df = pd.DataFrame({
    'Missing Values': missing_values_receipts,
    'Percentage of Total': missing_values_receipts_percentage
})

missing_values_items_df = pd.DataFrame({
    'Missing Values': missing_values_items,
    'Percentage of Total': missing_values_items_percentage
})

print("Missing values in receipts:\n", missing_values_receipts_df)
print("\nMissing values in items:\n", missing_values_items_df)
```

```
Missing values in receipts:
                          Missing Values  Percentage of Total
receipt_id                             0             0.000000
bonusPointsEarned                    575            51.385165
bonusPointsEarnedReason              575            51.385165
createDate                             0             0.000000
dateScanned                            0             0.000000
finishedDate                         551            49.240393
modifyDate                             0             0.000000
pointsAwardedDate                    582            52.010724
pointsEarned                         510            45.576408
purchaseDate                         448            40.035746
purchasedItemCount                   484            43.252904
rewardsReceiptStatus                   0             0.000000
totalSpent                           435            38.873995
userId                                 0             0.000000

Missing values in items:
                                     Missing Values  Percentage of Tota
l
receipt_id                                        0             0.000000
barcode                                        3851            55.481919
brandCode                                      4341            62.541421
competitiveProduct                             6296            90.707391
competitorRewardsGroup                         6666            96.038035
deleted                                        6932            99.870336
description                                     381             5.489123
discountedItemPrice                            1172            16.885175
finalPrice                                      174             2.506843
itemNumber                                     6788            97.795707
itemPrice                                       174             2.506843
metabriteCampaignId                            6078            87.566633
needsFetchReview                               6128            88.286990
needsFetchReviewReason                         6722            96.844835
originalFinalPrice                             6932            99.870336
originalMetaBriteBarcode                       6870            98.977093
originalMetaBriteDescription                   6931            99.855929
originalMetaBriteItemPrice                     6932            99.870336
originalMetaBriteQuantityPurchased             6926            99.783893
originalReceiptItemText                        1181            17.014839
partnerItemId                                     0             0.000000
pointsEarned                                   6014            86.644576
pointsNotAwardedReason                         6601            95.101570
pointsPayerId                                  5674            81.746146
preventTargetGapPoints                         6583            94.842242
priceAfterCoupon                               5985            86.226768
quantityPurchased                               174             2.506843
rewardsGroup                                   5210            75.061230
rewardsProductPartnerId                        4672            67.310186
targetPrice                                    6563            94.554099
userFlaggedBarcode                             6604            95.144792
userFlaggedDescription                         6736            97.046535
userFlaggedNewItem                             6618            95.346492
userFlaggedPrice                               6642            95.692263
userFlaggedQuantity                            6642            95.692263
```

1. bonusPointsEarned & bonusPointsEarnedReason: Over 51% of the records are missing these fields, making it difficult to analyze the reward points earned and the reasons behind them.
2. finishedDate & pointsAwardedDate: Nearly 50% of the records lack these dates, impacting time-based analyses related to the completion and awarding of points.
3. pointsEarned: Missing in approximately 46% of records, which affects the accuracy of reward points analysis.
4. purchaseDate: Absent in 40% of the records, complicating the tracking of purchase trends and patterns over time.
5. purchasedItemCount: Missing in 43% of the records, affecting inventory management and sales analysis.
6. totalSpent: Absent in nearly 39% of the records, leading to potential inaccuracies in financial analysis and spending patterns.

In [29]:
```

data_types_receipts = receipts_df.dtypes
data_types_items = items_df.dtypes
print(f"\nData types in receipts:\n{data_types_receipts}")
print(f"\nData types in items:\n{data_types_items}")
```

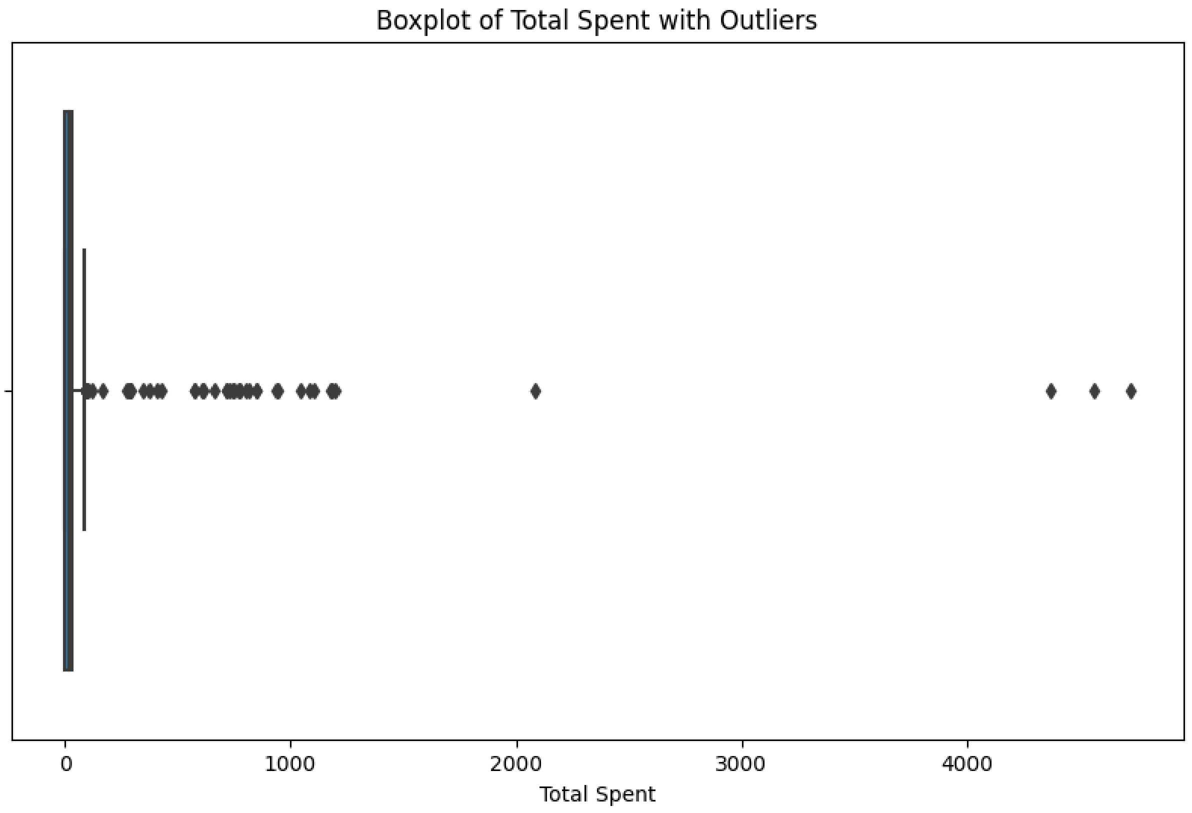
```
Data types in receipts:
receipt_id                         object
bonusPointsEarned                 float64
bonusPointsEarnedReason            object
createDate                 datetime64[ns]
dateScanned                datetime64[ns]
finishedDate               datetime64[ns]
modifyDate                 datetime64[ns]
pointsAwardedDate          datetime64[ns]
pointsEarned                       object
purchaseDate               datetime64[ns]
purchasedItemCount                float64
rewardsReceiptStatus               object
totalSpent                         object
userId                             object
dtype: object

Data types in items:
receipt_id                             object
barcode                                object
brandCode                              object
competitiveProduct                     object
competitorRewardsGroup                 object
deleted                                object
description                            object
discountedItemPrice                    object
finalPrice                             object
itemNumber                             object
itemPrice                              object
metabriteCampaignId                    object
needsFetchReview                       object
needsFetchReviewReason                 object
originalFinalPrice                     object
originalMetaBriteBarcode               object
originalMetaBriteDescription           object
originalMetaBriteItemPrice             object
originalMetaBriteQuantityPurchased    float64
originalReceiptItemText                object
partnerItemId                          object
pointsEarned                           object
pointsNotAwardedReason                 object
pointsPayerId                          object
preventTargetGapPoints                 object
priceAfterCoupon                       object
quantityPurchased                     float64
rewardsGroup                           object
rewardsProductPartnerId                object
targetPrice                            object
userFlaggedBarcode                     object
userFlaggedDescription                 object
userFlaggedNewItem                     object
userFlaggedPrice                       object
userFlaggedQuantity                   float64
dtype: object
```

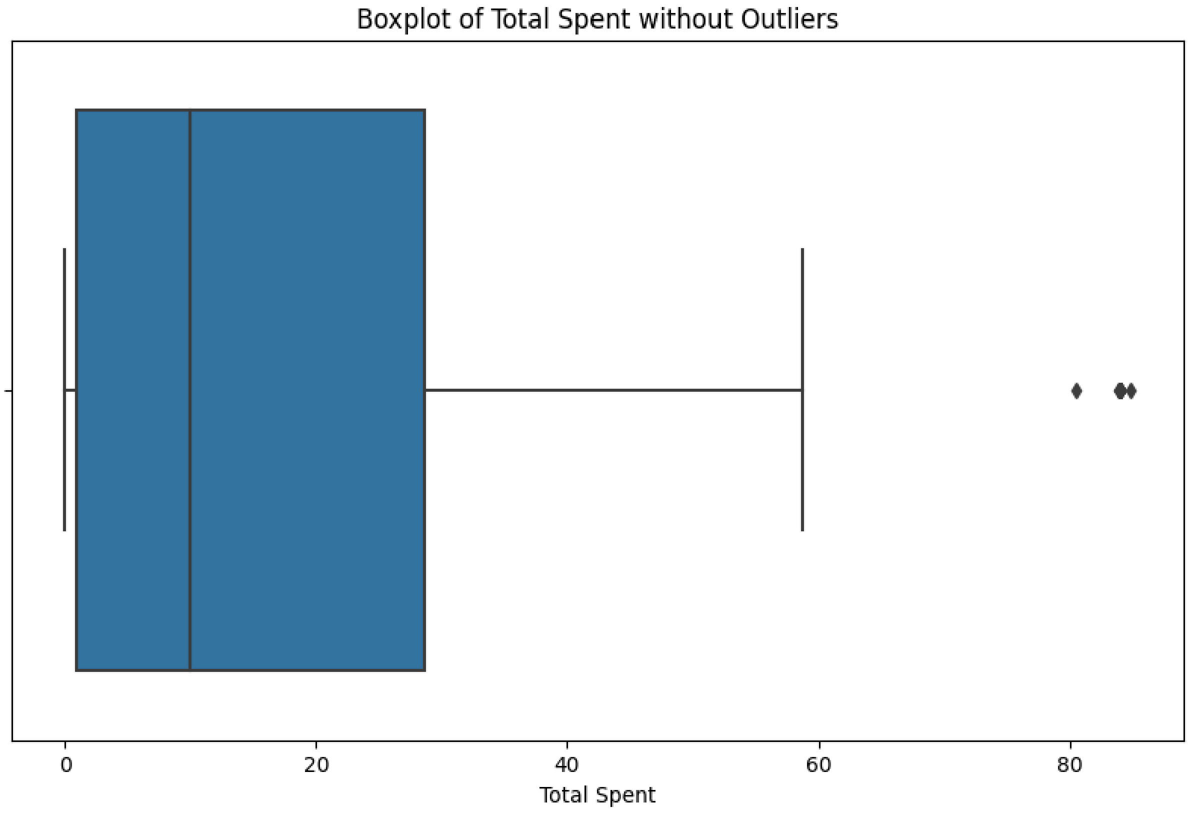
In [30]:
```
unique_statuses = receipts_df['rewardsReceiptStatus'].unique()
print(f"\nUnique values in 'rewardsReceiptStatus': {unique_statuses}")
```

```

Unique values in 'rewardsReceiptStatus': ['FINISHED' 'REJECTED' 'FLAGGE
D' 'SUBMITTED' 'PENDING']
```

In [40]:
```


def detect_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    outliers = df[(df[column] < (Q1 - 1.5 * IQR)) | (df[column] > (Q3 + 
    return outliers


receipts_df = pd.DataFrame(receipts_list)
receipts_df['totalSpent'] = pd.to_numeric(receipts_df['totalSpent'], err

outliers_totalSpent = detect_outliers(receipts_df, 'totalSpent')
print(f"Number of outliers in 'totalSpent': {outliers_totalSpent.shape[0


plt.figure(figsize=(10, 6))
sns.boxplot(x=receipts_df['totalSpent'])
plt.title('Boxplot of Total Spent with Outliers')
plt.xlabel('Total Spent')
plt.show()


receipts_no_outliers = receipts_df[~receipts_df.index.isin(outliers_tota


plt.figure(figsize=(10, 6))
sns.boxplot(x=receipts_no_outliers['totalSpent'])
plt.title('Boxplot of Total Spent without Outliers')
plt.xlabel('Total Spent')
plt.show()
```

```
Number of outliers in 'totalSpent': 55
```



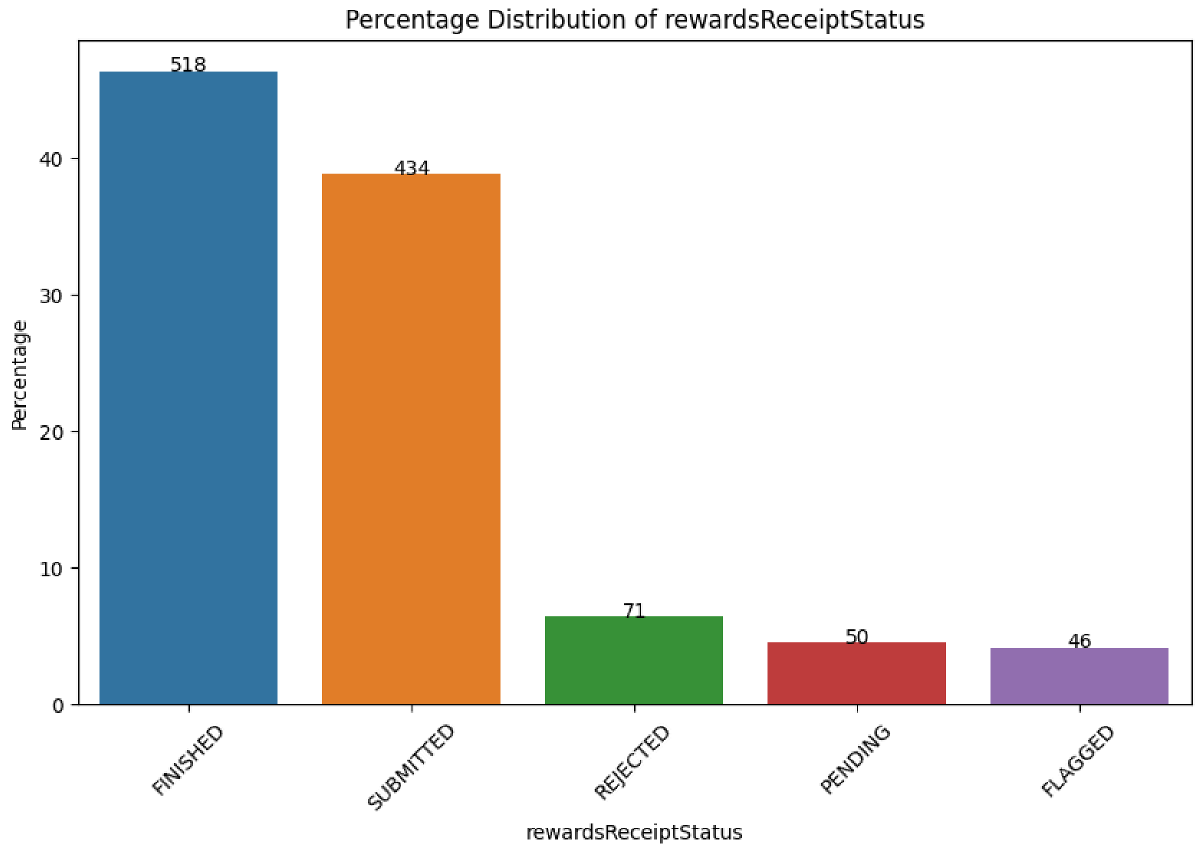


1. The boxplot with outliers shows several extreme values far beyond the main cluster of data. These extreme outliers extend beyond 1000, 2000, and even 4000 units.
2. Unusually high values in totalSpent may indicate potential fraudulent transactions that need further investigation.

3. These unusually high totalSpent amount can be indicative of some luxury purchases that

In [32]:

```python

def plot_percentage_bar_chart(column_name, data_frame):
    total_count = len(data_frame)
    value_counts = data_frame[column_name].value_counts()
    percentages = (value_counts / total_count) * 100

    plt.figure(figsize=(10, 6))
    sns.barplot(x=percentages.index, y=percentages.values)
    plt.title(f'Percentage Distribution of {column_name}')
    plt.xlabel(column_name)
    plt.ylabel('Percentage')
    plt.xticks(rotation=45)


    for index, value in enumerate(value_counts):
        plt.text(index, percentages.values[index], f'{value}', ha='cente

    plt.show()


plot_percentage_bar_chart('rewardsReceiptStatus', receipts_df)
```



The Data quality issues I found in the receipts.json-

1. Data related to flagged items in the receipts JSON is inconsistent.
2. Attributes like userFetchedReview and needsFetchedReview indicate uncertainty associated with the review of the purchase.

3. Including uncertain and incomplete data could compromise the integrity and reliability of the ER model therefore these attributes are excluded to maintain a clean, efficient, and trustworthy data model until data quality issues are resolved.

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```